

Research Direction: Reservoirs as Dynamic LSH for CCSLM-Based Reinforcement Learning

Core idea

Explore the combination of **reservoir computing**, **dynamic locality-sensitive hashing**, and **CCSLM** as a potentially simple architecture for reinforcement learning.

The key observation is:

- A reservoir provides a **dynamic state representation**:
 - current observation
 - previous observations
 - recent actions
 - temporal history
- Random projection of the reservoir state followed by binarization produces a **dynamic LSH**:
 - $r(t) \rightarrow \text{sign}(H r(t))$
- The resulting binary context can control CCSLM:
 - $y(t) = L(C(t), x(t))$
- Thus the system becomes a **history-dependent selection of linear mappings**.

Why this may matter for RL

- RL frequently has useful information distributed across time.
- A current observation may not contain sufficient information to determine the appropriate action.
- A reservoir provides temporal memory without requiring the downstream computation itself to be recurrent.
- Binary reservoir states provide a compact, high-dimensional **coordinate system for temporal context**.
- CCSLM can learn different linear mappings in different regions of that context space.

Potential architecture

- **Data pathway**
 - observation/features $\rightarrow$ CCSLM $\rightarrow$ policy/value output
- **Control pathway**
 - observation + previous state/action $\rightarrow$ reservoir
 - reservoir state $\rightarrow$ random projection
 - random projection $\rightarrow$ binary context
 - binary context $\rightarrow$ selects/modulates CCSLM mappings

This preserves the CCSLM principle:

Separate data from control.

Interesting interpretation

The reservoir can be viewed less as a conventional neural network and more as:

- a **dynamic random projection**
- a **temporal feature generator**
- a **history-dependent locality-sensitive hash**
- a mapping from trajectories into computational state space

The CCSLM then performs:

- context-dependent linear prediction
- local specialization
- temporal conditional computation
- potentially hierarchical refinement of the context space

Research questions

- How much RL performance can be obtained with a **fixed random reservoir**?
- Does binarizing the reservoir preserve the useful temporal information?
- How does the number of context bits affect policy/value approximation?
- Do similar trajectories produce useful LSH neighborhoods?
- Do CCSLM mappings naturally cluster according to behavioral situations?
- Can the **Context Specialization Metric** identify useful versus redundant reservoir bits?
- How does this compare with conventional recurrent networks and transformers?
- Can different CCSLM stages use different temporal contexts?
- Can the reservoir itself remain completely untrained?

Particularly attractive experiment

Start with a small partially observable RL environment.

Compare:

- conventional MLP
- recurrent network
- fixed reservoir + linear readout
- fixed reservoir + binary LSH + CCSLM
- fixed reservoir + binary LSH + multiple CCSLM stages

Measure:

- reward
- sample efficiency
- parameter count
- training stability
- inference cost
- memory requirements
- sensitivity to reservoir size and number of LSH bits

Hypothesis

A useful hypothesis is:

A fixed dynamic random projection can provide the temporal coordinate system, while CCSLM learns the computation required in each region of that space.

If successful, this would offer an unusual decomposition of RL:

temporal state generation → dynamic hashing → context-selected linear computation → policy/value

rather than requiring a large monolithic learned nonlinear system.

Broader possibility

This could provide a bridge between:

- reservoir computing
- locality-sensitive hashing
- associative memory
- switched linear systems
- CCSLM
- partially observable RL

The potentially important idea is not simply *using a reservoir in RL*, but treating the **binarized reservoir state as a dynamic LSH of trajectory history** and using that hash as the control plane for CCSLM.